Chapter 4

TSO and Database Research and Ad Hoc Queries

Contents

Chapter 4: TSO and Database Research and Ad Hoc Queries

Figures

4.1 NSLDS User Authorization

NSLDS contains loan-level information on students and institutions participating in the Title IV aid program that is subject to the restrictions of the Privacy Act of 1974. Access to, use, and disclosure of this information are controlled accordingly.

All users of NSLDS must be authorized by ED. Procedures for obtaining authorization to use the system are covered in Chapter 2 of this User Documentation. For ready reference, those procedures are summarized here:

1. You must be a member of a participating organization, such as ED, another Government organization or agency, or a school, guaranty agency (GA), or servicer approved by ED.

2. You must apply to ED, NSLDS Division, for a personal user ID.

3. You must hold the appropriate security clearance and have a need-to-know reason for accessing the data.

4. If you are a non-ED user, you must first be registered with the Title IV Wide Area Network (WAN) as your means of communication to the system and hold a valid Title IV WAN ID number before you can apply for an NSLDS account.

5. You must comply with the provisions of the Privacy Act of 1974 (as amended), and sign, or have previously signed during the security clearance process, a Privacy Act Acknowledgment Statement indicating your awareness and understanding of the Privacy Act restrictions and your agreement to comply with those restrictions.

4.2 Hardware and Software Requirements

The NSLDS central database is housed on a contractor mainframe at the NSLDS Data Center. You can access the system through your local workstation and the telecommunications media for which you are authorized. In order for you to access and use the NSLDS, your workstation must meet the following hardware and software requirements:

- **Terminal**—A device capable of IBM 3270 emulation, including IBM-compatible personal computers (PCs), Apple computers, terminals, and similar devices.

- **RAM**—A minimum 512K Random Access Memory (RAM).

- **Hard Drive**—Space requirements depend on the size of the files you download to your PC. When calculating the space needed to download a file, be sure to consider the amount of space used on your hard drive by your other applications.

- **Disk Drive**—A 3.5-inch floppy disk drive, either double density (DD) or high density (HD).

- **Modem**—A fully Hayes-compatible asynchronous modem capable of a 14.4 or 28.8 baud exchange rate.

- **Operating System**—IBM or MS-DOS, Version 3.3 or later.

- **Software**—3270 terminal emulation communications package, such as IBM Communication Manager, AttachMate, ProComm, or CrossTalk.

4.3 Logging on and Accessing NSLDS

NSLDS is normally available for online transactions at all times except the following maintenance periods. All times are Central time.

- Every week from 6:00 p.m. Saturday until 2 a.m. Sunday.
- Every week from 4:00 p.m. Sunday until 3:30 a.m. Monday.
- Every "off" week from 7:00 p.m. until 1:00 a.m., Monday through Friday. "Off" weeks occur every 13th week.

If maintenance or upgrade activities require that the system go offline outside these normal maintenance windows, the user community will be notified in advance.

There are several ways to access the NSLDS. Telecommunications routes for each access method are displayed in Figure 4–1. The following subsections provide instructions for using each of these methods.

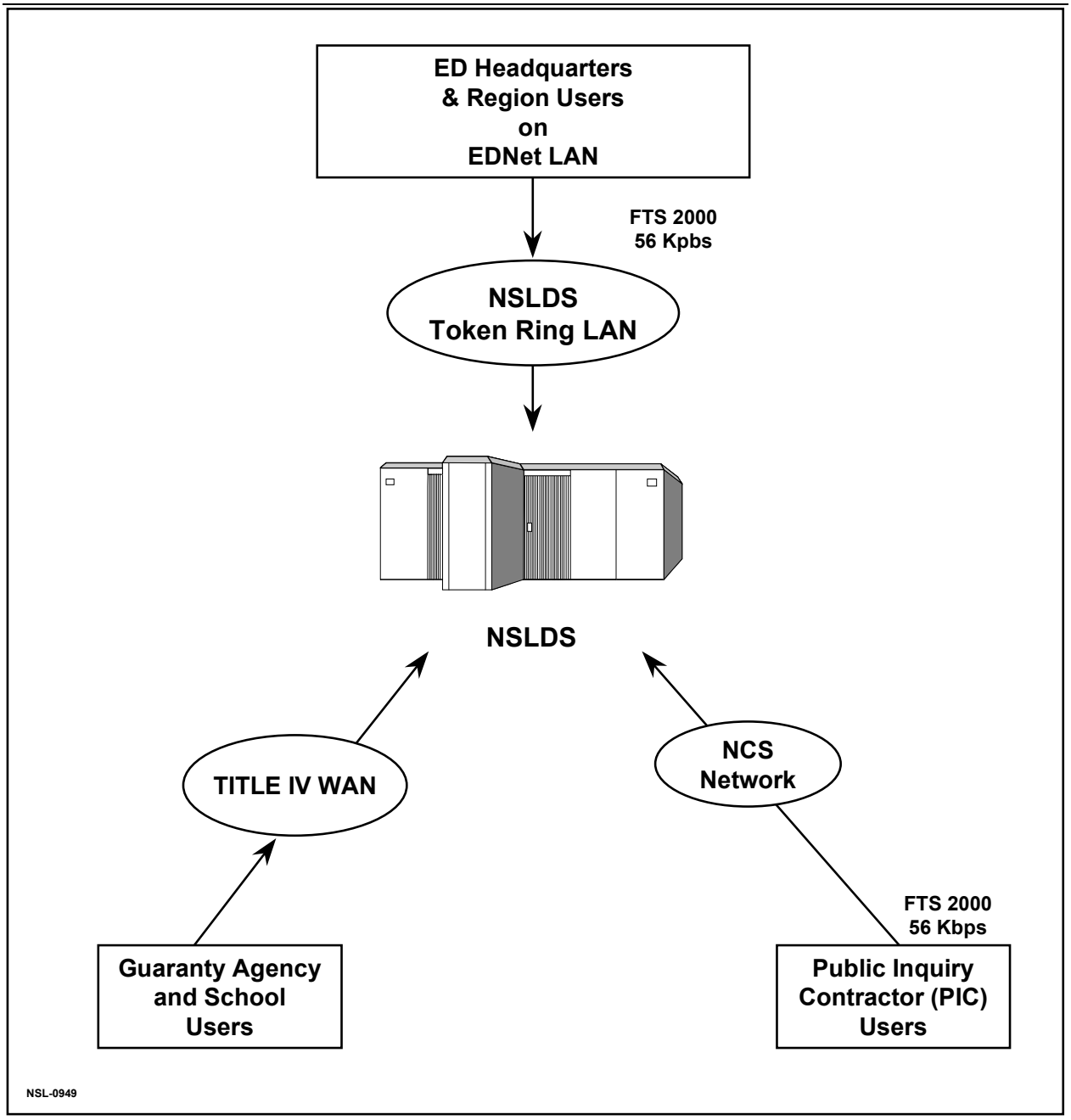

Figure 4–1, NSLDS Database Access

4.3.1 NSLDS User Logon

4.3.1.1 ED Users on the EDNet LAN

ED users at Department Headquarters and regional offices may access the NSLDS through ED's Local Area Network (EDNet LAN) by selecting the AttachMate icon (Figure 4–2). The icon

depicts a racing flag symbol for NSLDS that, when selected, acts as a hot key for automatically accessing the NSLDS mainframe.

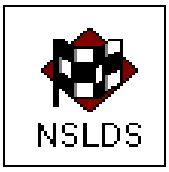

Figure 4–2, AttachMate Icon

Follow these steps to log on to the NSLDS from ED headquarters or regional offices:

1. Log on to the EDNet LAN following normal access procedures.

2. Double-click on the AttachMate icon that appears on your desktop or Start menu to automatically access the NSLDS mainframe. The *Welcome to NSL* menu (Figure 4–3) displays.

```
 WELCOME TO NSL SYSTEM - CSC MERIDEN - OS/390

********************************************************************************
* THIS SYSTEM IS LIMITED TO ACCESS BY AUTHORIZED PERSONNEL, FOR APPROVED USES *
* ONLY. ACCESS BY OTHERS IS PROHIBITED AND UNAUTHORIZED.                       *
********************************************************************************

 ENTER DESIRED OPTION:


 A. TSO
 B. NASN
 C. CICSNSLP
 D. CICSNSLT
 E. NSLPCSM
 F. OMVTAMN
 G. OM2NMVS
```

Figure 4–3, Welcome to NSL System Menu

Follow the steps outlined in Section 4.3.2 to log on to TSO.

4.3.1.2 Guaranty Agency and School User Access

GA and school users access the NSLDS mainframe through the Title IV WAN. This network connects users with ED and its contractors, and allows them to communicate directly with the NSLDS database.

Users who sign up for a Title IV WAN account are assigned a "mailbox" that is used both to request and to receive data from NSLDS. They also receive a WAN ID account number that serves as the mailbox identifier.

GA and school users access NSLDS online functions via NET*CONNECT and batch files via EDConnect. The Title IV WAN contractor provides schools and GAs with documentation for both these software products that explains how they can be used to log on to NSLDS.

4.3.1.3 Public Inquiry Contractor (PIC) Users

PIC users access the NSLDS mainframe through the contractor network from the *NCS Information Services Division* menu. When the menu appears, the cursor moves automatically to a Choice Entry field in the lower left portion of the screen. To access NSLDS from the *NCS Information Services Division* menu, follow these steps:

1. Select NSLDS access by typing ***PICNSL*** in the Choice Entry field (Figure 4–4). You are not restricted to the four options listed on the screen.

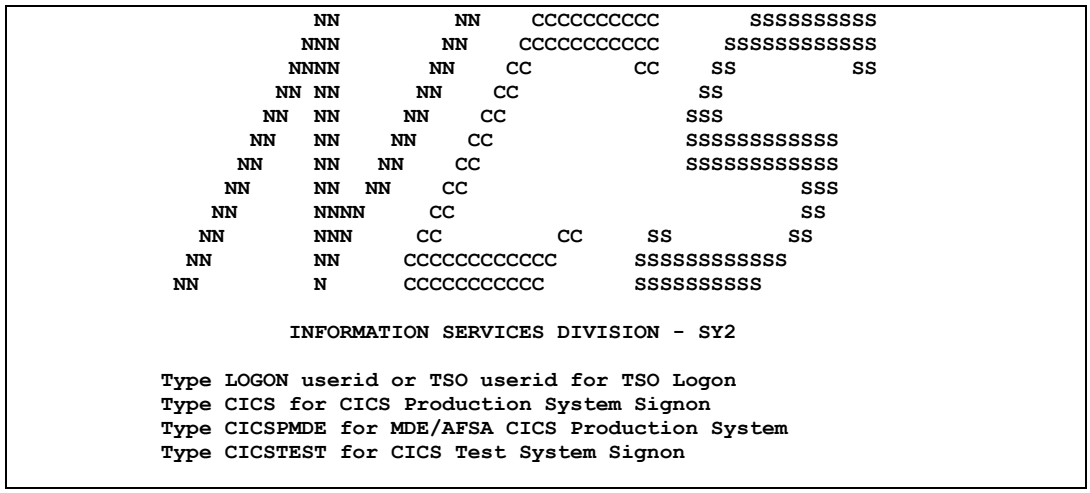

Figure 4–4, NCS Information Services Division Menu

2. Press **ENTER**. The system displays the *CICS Logon* screen (Figure 4–5).

```
ESGD INFORMATION SERVICES     CICSNSLP SYSTEM
     USERID:
     PASSWORD:        NEW PASSWORD (IF DESIRED):

DISCLOSURE STATEMENT
    The user understands that the Department of Education, its agents and
    sub-contractors have signed up to meet the requirements of the PRIVACY ACT OF
    1974 (as amended).  As such, by entering this system, the user hereby verifies
    that they have read the PRIVACY ACT OF 1974 (as amended), that the user
    understands the requirements of the ACT, and that the user has no remaining
    questions.

MONITORING NOTICE
    This system is for the user of authorized users only.  Individuals using this
    computer system without authority are subject to having all of their
    activities on this system monitored and recorded by system personnel.  In the
    course of monitoring individuals improperly using this system, or in the
    course of system maintenance, the activities of authorized users may also be
    monitored.  Anyone using this system expressly consents to such monitoring and
    is advised that if such monitoring reveals possible evidence of criminal
    activity, system personnel may provide the evidence of such monitoring to law
    enforcement officials.
```

Figure 4–5, CICS Logon Screen

4.3.2 TSO Logon

Different users access NSLDS applications from different screens, as described in Section 4.3.1. However, most ED users follow the steps described in Section 4.3.1.1 to access NSLDS from the EDNet LAN. Once they do so, they see the *Welcome to NSL System* menu (Figure 4–6).

```
 WELCOME TO NSL SYSTEM - CSC MERIDEN - OS/390

*******************************************************************************
* THIS SYSTEM IS LIMITED TO ACCESS BY AUTHORIZED PERSONNEL, FOR APPROVED USES *
* ONLY. ACCESS BY OTHERS IS PROHIBITED AND UNAUTHORIZED.                      *
*******************************************************************************

 ENTER DESIRED OPTION:


 A. TSO
 B. NASN
 C. CICSNSLP
 D. CICSNSLT
 E. NSLPCSM
 F. OMVTAMN
 G. OM2NMVS
```

Figure 4–6, Welcome to NSL System Menu

Of the seven options listed by the *Welcome to NSL System* menu, only one is available to ordinary users:

- **TSO**—Time Sharing Option (Option **A**). This option provides access to the Query Management Facility (QMF), Data File Download function, Report Management and Distribution System (RMDS), and a variety of other advanced functions.

When the *Welcome to NSL System* menu appears, your cursor moves automatically to an unmarked Choice Entry field beneath the list of available options. To log on to the TSO environment, follow the steps outlined in the subsections that follow.

4.3.2.1 TSO Logon

To log on to TSO from the *Welcome to NSL System* menu (Figure 4–6), follow these steps:

1. Select the TSO option by typing ***A*** in the Choice Entry field.

2. Press **ENTER**.

IBM 3270 Terminal Users: When you are instructed to press ENTER, press the RIGHT CTRL key on your keyboard as the equivalent for ENTER. If you are using different terminal emulation software, check your software documentation for information on keyboard mapping and how it determines which keys you press to perform specific actions.

3. The system displays an "ENTER USERID" prompt (Figure 4–7), and the cursor automatically advances to a Data Entry field beneath the prompt.

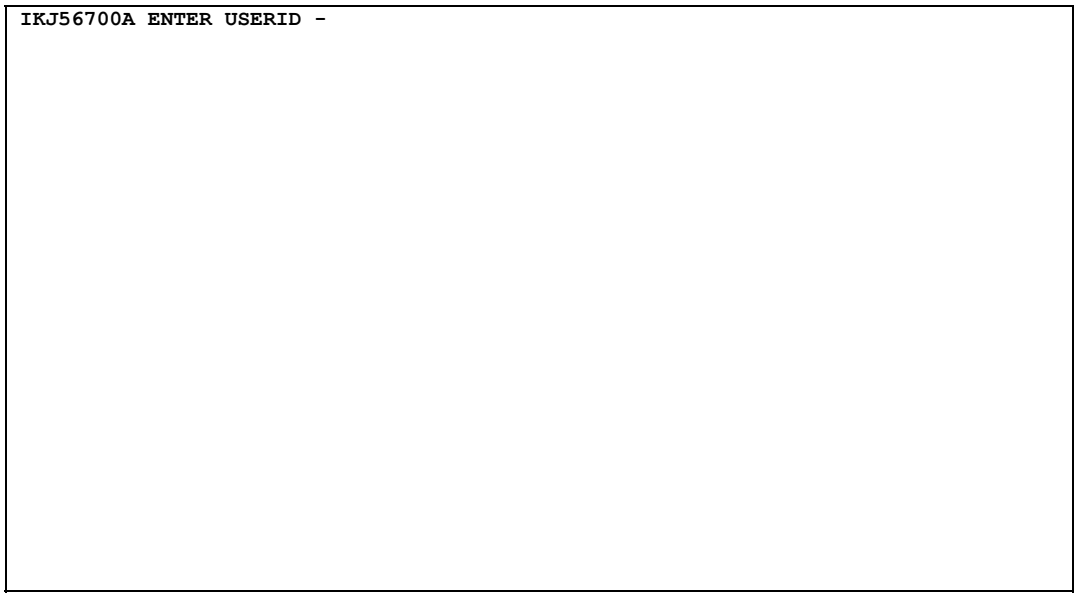

Figure 4–7, ENTER USERID Prompt

4. Type your **user ID** in the Data Entry field and press **ENTER**.

5. The *TSO/E Logon* screen (Figure 4–8) displays. Your user ID is automatically displayed in the Userid field, and the cursor advances to the Password field.

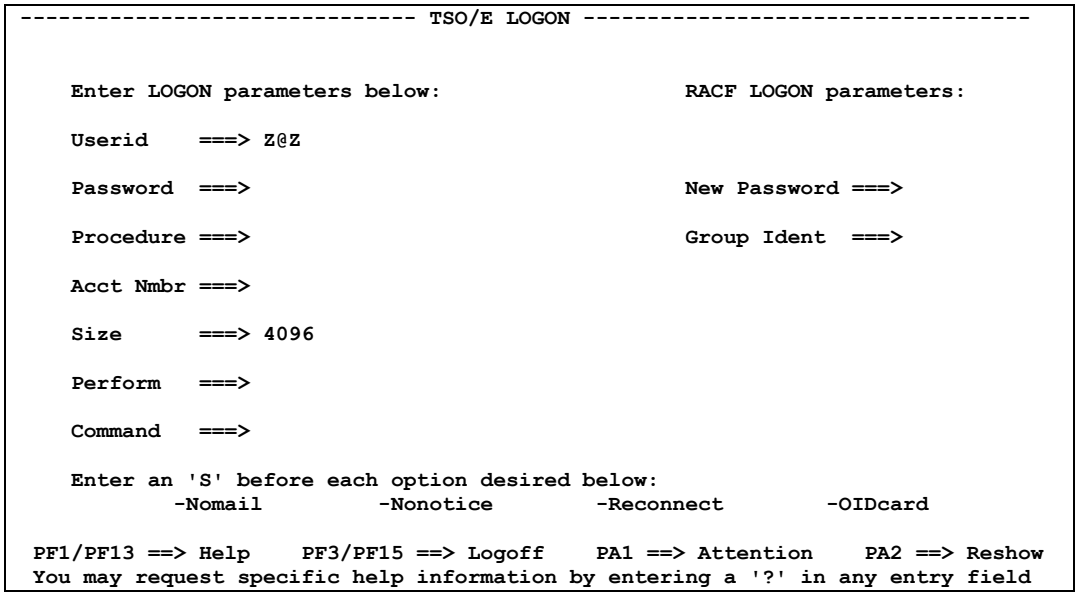

Figure 4–8, TSO/E Logon Screen

6. Type your **Password** in the Password field and press **ENTER**. Your password is not displayed as you type, but the cursor advances across the screen. The only information you must type on this screen is your password.

When you log on to TSO for the first time, your initial password expires. The system displays the following message near the top of the screen, "IKJ56415I CURRENT PASSWORD HAS EXPIRED—PLEASE ENTER NEW PASSWORD" (Figure 4–9). You must change your password at this time (see Section 4.3.2.1.1 and Section 4.3.2.1.2).

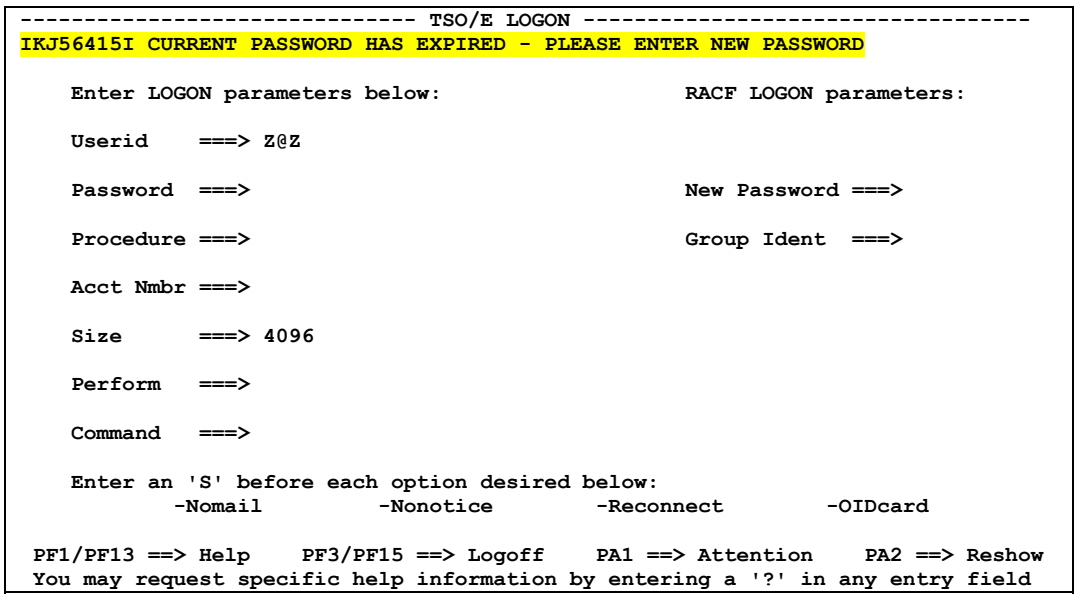

Figure 4–9, Current Password Expired Message

4.3.2.1.1 Changing Your Password in TSO

To change your password in TSO, follow these steps:

1. Type your **New Password** in the New Password field. Your password is not displayed as you type, but the cursor advances across the screen.

2. Press **ENTER**. The system displays the following message, "IKJ56447A Reenter the new password in the NEW PASSWORD field for verification", and the cursor remains in the New Password field (Figure 4–10).

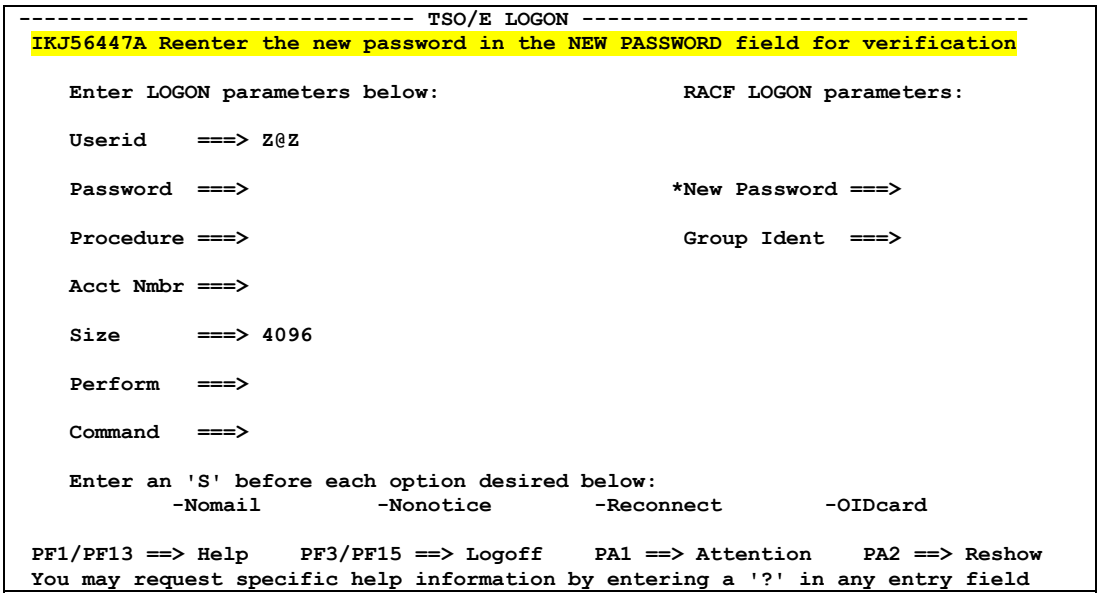

Figure 4–10, Reenter New Password Message

3. Retype your **New Password** exactly as you did before and press **ENTER**. Your password is not displayed as you type, but the cursor advances across the screen.

4.3.2.1.2 Invalid User ID or Password in TSO

If you have entered an invalid user ID or password, the system displays an informational message to that effect at the top of the screen (Figure 4–11).

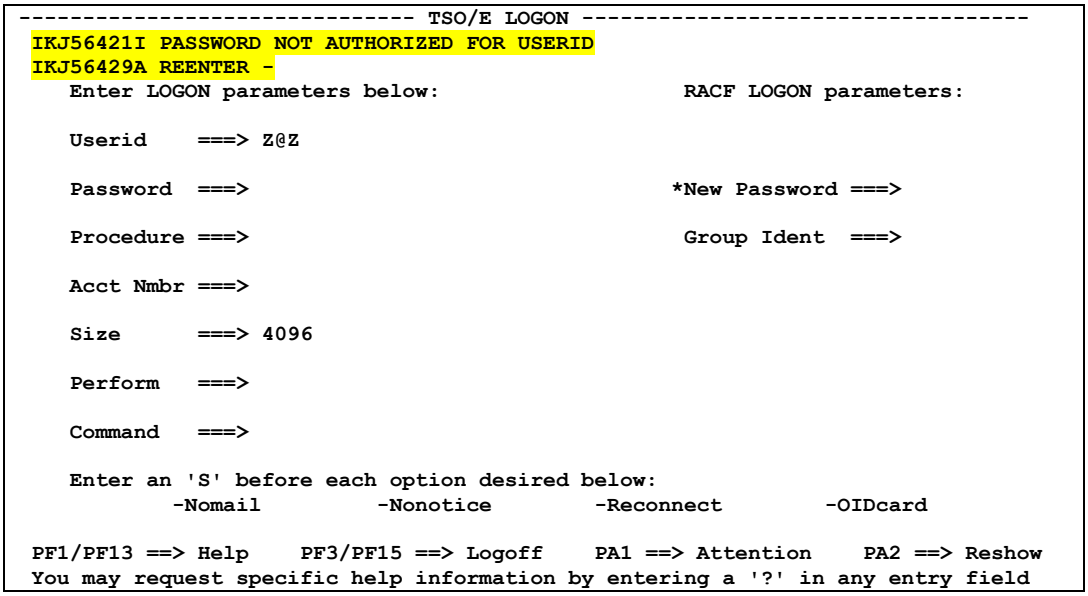

Figure 4–11, Invalid User ID Message

After three unsuccessful logon attempts, the system activates a lockout and displays an "Invalid Sign-on attempt" message if you try to log on to TSO. In this event, you must contact the CSC at 1-800-999-8219 and request that your user ID and password be reset. To log on again, you must repeat the steps in Section 4.3.2.1.1.

4.3.2.1.3 Successful Logon to TSO

Upon successful logon, the *NSLDS Disclosure Statement and Monitoring Notice* screen displays. The Disclosure Statement and Monitoring Notice are shown in Figure 4–12 as if they occupied a single screen. In fact, because of space constraints, they are spread across two screens. You must press **ENTER** to move from the first screen to the second screen.

All information in NSLDS is protected by Federal regulations under the Privacy Act of 1974 (as amended). You should familiarize yourself with these regulations.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *                                                                                     *
 *            YOU ARE LOGGED ONTO NSLDS TSO AT CSC IN MERIDEN, CT                      *
 *                                                                                     *
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

  Questions should be directed to the NSLDS Customer Service Center at
  800.999.8219

                                Disclosure Statement

 The user understands that the Department of Education, its agents and
 Sub-contractors have signed up to meet the requirements of the "PRIVACY ACT of
 1974" (as amended).  As such, by entering this system, the user hereby verifies
 that he/she has read the "PRIVACY ACT of 1974" (as amended), that the user
 understands the requirements of the act, and that the user has no remaining
 unanswered questions.

                                 Monitoring Notice

 This system is for the use of authorized users only.  Individuals using this
 Computer system without authority, are subject to having all of their
 Activities on this system monitored and recorded by system personnel.  In the
 Course of monitoring individuals improperly using this system or in the course
 Of system maintenance, the activities of authorized users may also be
 Monitored.  Anyone using this system expressly consents to such monitoring and
 Is advised that if such monitoring reveals possible evidence of criminal
 Activity, system personnel may provide the evidence of such monitoring to law
 Enforcement officials.
 ***
```

Figure 4–12, Disclosure Statement and Monitoring Notice

Press **ENTER** to continue. When you have successfully logged on to TSO, the system displays the *NSLDS—ISPF/PDF Primary Option Menu* screen (Figure 4–13).

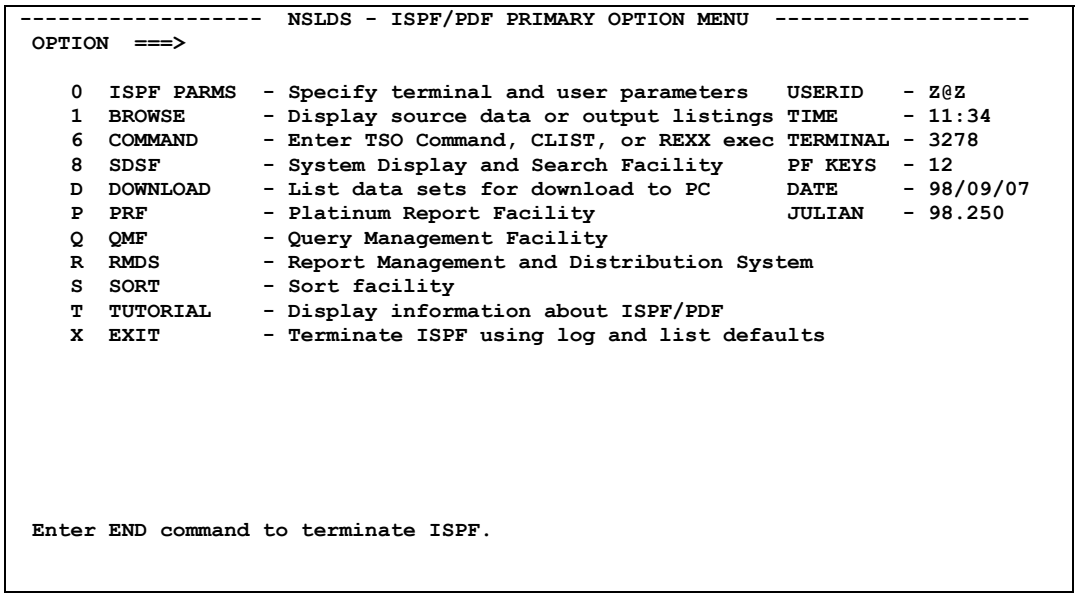

Figure 4–13, NSLDS—ISPF/PDF Primary Option Menu

Note: Option **P** (Platinum Report Facility) is no longer available to ED. Use QMF instead.

4.3.2.1.4 NSLDS Primary Option Menu

Some of the options listed on the *NSLDS—ISPF/PDF Primary Option Menu* screen are described below. Your authorization level determines which tools are available to you.

- **ISPF PARMS**—(Option **0**). This option allows you to set various parameters that affect how your workstation displays information or performs actions. For example, you may choose whether to display function keys at the bottom of your screen. Function keys provide screen display and navigation and serve as shortcuts for invoking TSO commands. These commands are listed in the online tutorial (Option **T**). Within TSO, you can access online help anytime by pressing **F1**.

- **BROWSE**—(Option **1**). This option allows you to display source data or output listings in a full-screen format and to scroll through the display for easy access.

- **COMMAND**—(Option **6**). This option gives you access to a command line from which to run executable programs.

- **SDSF**—System Display and Search Facility (Option **8**). This option allows you to monitor and control jobs, queues, and system resources such as printers.

- **Download**—(Option **D**). This option allows you to download a report or extract from the mainframe to your PC.

- **QMF**—Query Management Facility (Option **Q**). This option allows you to query the NSLDS databases on an ad hoc basis. QMF queries can be created, saved, and executed on the NSLDS database.

- **RMDS**—Report Management and Distribution System (Option **R**). This option allows you to browse reports online, share them online with other authorized users, print all or selected pages of a report, or route all or selected pages of a report to diskette, magnetic tape, Title IV WAN, or a data set for download to a PC.

- **SORT**—Sort Facility (Option **S**). This option allows you to select data sets and build control statements needed for the SORT facility or equivalent programs.

The cursor advances to the Option prompt at the top left of the screen. Refer to the following chapters for detailed information about how to use these functions:

- Chapter 5—Query Management Facility
- Chapter 7—Download
- Chapter 8—Report Management and Distribution System

To select the option you seek from the *NSLDS—ISPF/PDF Primary Option Menu* screen, follow these steps:

1. Type ***1, 8, D, P, Q, R,*** or ***S*** at the "Option" prompt.
2. Press **ENTER**. The corresponding screen displays.
3. Press **F3** to return to the NSLDS—ISPF/PDF Primary Option Menu screen

4.4 Introduction to Database Research

This chapter introduces the NSLDS database, its logical design and physical characteristics, and methods for accessing the NSLDS data using ad hoc queries. Topics introduced in this chapter include:

- Database Structure
- Organization of Data
- Available Data Tables
- Definition of Table Attributes
- Keys and Indexes for Accessing Data

The previous chapters explain how to use NSLDS online and batch procedures. Those procedures answer most of the routine needs of the Title IV user community. For users requiring specific, tailored information that cannot be accumulated using standard online and batch procedures, NSLDS also provides access to the database by means of custom, ad hoc queries.

This chapter will teach you how to access the database, collect customized information, and create your own reports where standard reports do not exist.

Even if you are a beginner at using ad hoc queries for research, you can quickly learn about the databases, tables, and columns (attributes) that make up NSLDS, and soon use this chapter for reference only.

Before creating a custom design, be sure your need cannot be satisfied with a standard report. Custom, ad hoc queries are an unnecessary expense if a standard report is available.

Save your queries and run them as many times as needed. If you find you are frequently running the same query with little or no variation, consider having your report become a system standard. Pre-programmed reports are significantly more efficient and less costly to run.

4.4.1 NSLDS—Two Databases in One

NSLDS maintains an Active database and an Online Abstract database, each of which contains many tables. To access either of these databases, follow these steps:

- Log on using the Time Sharing Option (TSO) procedure described earlier in this chapter.
- Select an operating environment from the *NSLDS—ISPF/PDF Primary Option Menu.*
- Select the Query Management Facility (QMF) ad hoc query tool. Chapter 6 describes how to use this query facility.

Use the table and attribute names provided in this chapter to access tables in the Active Database. To access the Online Abstract Database, type the characters ***OSAP.*** before the table name.

To total the Cumulative Principal Collection Amount of IRS Offsets by GA from the IRS Offset Table in the Online Abstract Database, you might formulate the following query:

```
SELECT      GA_CODE
            SUM(CUM_PRIN_COLL_AMT)
FROM        OSAP.IRS_OFF
GROUP BY    GA_CODE
```

To submit the same query to the Active Database, eliminate the ‘OSAP.’ in the third line of the query.

Do NOT mix table and attribute names from multiple databases in the same query. This action could produce MEANINGLESS or MISLEADING RESULTS.

4.4.1.1 Active Database

The Active Database contains loan-level data for student accounts. An account contains all loan and grant records associated with a single student. A student account is considered open until the following conditions are met:

- All loans within the account have had a *closed loan* status for at least 15 months. Codes for *closed loan status* are defined in Figure 4–14.
- The *date of maturity* for every loan in the account is at least six years earlier than the *current date.*
- The student owes no grant overpayments.

Closed Status	
Code	**Reason**
BC	Bankruptcy Claim, Discharged
CA	Canceled
CS	Closed School Discharge
DC	Defaulted, Compromise
DD	Defaulted, Then Died
DE	Death
DI	Disability
DK	Defaulted, Then Bankrupt, Discharged, Chapter 13
DN	Defaulted, Then Paid in Full Through Consolidation Loan
DP	Defaulted, Paid in Full
DR	Defaulted Loan Included in Roll-Up Loan
DS	Defaulted, Then Disabled
DW	Defaulted, Write-Off
FC	False Certification Discharge
OD	Defaulted, Then Bankrupt, Discharged, Other

Closed Status	
Code	**Reason**
PC	Paid in Full Through Consolidation Loan
PF	Paid in Full
PM	Presumed Paid in Full
PN	Non-Defaulted, Then Paid in Full Through Consolidation Loan
RF	Refinanced
UC	Permanently Uninsured/Unreinsured, Default Claim Denied
UD	Permanently Uninsured/Unreinsured, No Default Claim Requested
UI	Uninsured/Unreinsured

Figure 4–14, Closed Loan Status Codes

The user-accessible tables in the NSLDS database are grouped into seven subject areas:

1. Schools
2. Students (includes Grants)
3. Guaranty Agencies
4. Lenders
5. FDLP Servicers
6. Loans
7. Default Rates

The tables that make up the Active Database are listed in Section 4.4.3, and the column (attribute) definitions for each table are listed in Section 4.4.4.

4.4.1.2 Online Abstract Database

The large number of loans and grants issued under Title IV student aid programs inevitably suggests using a subset of the NSLDS database for statistical and research purposes. The Online Abstract Database is a small subset of the NSLDS database containing loan, grant, and student data for a statistically valid, random sampling of 1.5 million borrowers selected from the Active Database. The tables that make up Online Abstract Database are listed in Section 4.4.3, and the column (attribute) definitions for each table are listed in Section 4.4.4. These tables can be accessed by typing ***OSAP.*** in front of their names.

4.4.2 How NSLDS Data Is Organized for Ease of Use

The NSLDS database was built with a relational database tool. In a relational database, data is organized into tables consisting of rows and columns. Each table stores information about one entity type. An entity type can be a collection of people, places, things, or events that share

common definitions, relationships, and attributes. Examples of Title IV entity types include schools, students, loans, and GAs.

The rows of a table in a relational database represent the specific instances (or entities in relational database terms) of an entity type. They are similar to records in a traditional file structure. The School table, for example, contains one row for each school eligible to participate in Title IV programs.

The columns (attributes) of a table contain data values that describe the entity. They are similar to fields or data elements in a traditional file structure record. The columns in the School table include the following:

- **ACT_FLAG**—A flag indicating whether an institution is eligible and certified to participate in Title IV program
- **CODE**—A six-digit ED code for uniquely identifying a school
- **NM**—The name of the school

Each intersection of a row and column is called a cell. The cell stores a value of the specific information defined by the intersection of row and column. The following are two examples:

Example 1: Date of Birth Cell—The Student table has a *row* for each person who is, or was, a student with Title IV aid, and a *column* for each attribute relating to individual students, such as name, Social Security Number, and date of birth. The cell that represents the intersection of the row for the student, John Doe, and the Date of Birth column might contain the value 1974-05-12 or May 12, 1974, as shown in Figure 4–15.

Title IV Aid Students		
Name of Student	**Date of Birth**	SSN
Alice Deer	March 20, 1949	123-45-6789
John Doe	**May 12, 1974**	234-56-7891
Walter Stag	December 7, 1969	345-67-8910

Figure 4–15, Date of Birth Cell

The attribute (or attributes) that uniquely identifies a row in a table is called the identifier or primary key. To select a specific row from a table, you must specify the identifier. The identifier representing a specific row must be different from every other identifier in that table.

Example 2: School Branch Identifier Cells—A School Branch is uniquely identified by its six-digit main campus identifier and its two-digit branch code in the School Branch table. The six-digit main campus code 777777 and the two-digit branch identifier 03, when combined as identifiers for a row, uniquely identify one institution. Other institutions can also be identified by the same main campus code 777777, but they must have a branch code other than 03 to co-exist in the same School Branch table, as shown in Figure 4–16.

School Branch		
School Code	**School Branch Code**	**School Branch Name**
777777	00	UNIVERSITY OF EDUCATION - MAIN BRANCH
777777	01	UNIVERSITY OF EDUCATION - BRANCH 01
777777	02	UNIVERSITY OF EDUCATION - BRANCH 02
777777	**03**	**UNIVERSITY OF EDUCATION - BRANCH 03**
777777	04	UNIVERSITY OF EDUCATION - BRANCH 04

Figure 4–16, School Branch Identifier Cells

In a relational database, information about related entities is often divided into separate tables. For instance, a PLUS borrower is related to one or more loans, but to store a date of birth with every PLUS loan would take up too much space. Instead, personal information such as the date of birth of a PLUS borrower is placed in one table, and all loan information for that borrower in another. The two are then connected, or joined, with a relationship.

The relational database environment supports relationships through linkages, called *joins*, on the attributes shared by tables. Related attributes are defined by Primary and Foreign Keys, which are discussed in Section 4.4.5. In ad hoc queries, you may query single or joined tables. In fact, joins are very important to in-depth research.

To form a valid join and relate data from specific rows in one table to specific rows in another table, you must make sure that the identifiers of one table appear as attributes in the other table. Before you can join a PLUS borrower with the PLUS borrower's loans, the identifier for the PLUS Borrower table (SSN) must exist as an attribute in the Loan table. Otherwise, it would be impossible to know which loans belong to a given borrower. To join tables using a QMF, be sure to utilize the related attributes from each table.

One point about efficiency: Two queries can produce the same results, but one may runs for an hour and the other for five minutes. What is the difference? Indexes. Some attributes have indexes, which are shortcuts to data, much the same as an index at the back of a book is a pointer to the page containing the information that is needed. Queries run much faster if you use attributes in the WHERE clauses that have pre-established indexes. For more information on indexes, see Section 5.1.6.

4.4.3 Specific NSLDS Tables You Can Access

Figure 4–1 lists the names and briefly describes the contents of the NSLDS tables users can access by means of ad hoc queries. An asterisk in the High Volume column indicates that a table

contains more than 5 million records and that queries run against it should be written with special care to avoid tying up system resources for an excessive amount of time.

Table Description	Table Name	High Volume	Database	
Accounting Calendar	ACCT_CALNDR		A	
Activity Function	ACT_FUNC		A	
Address Support	ADD_SUPT		A	
Aggregate Description	AGG_DESCN		A	
Aggregate Descriptor	AGG_DESC		A	
Aggregate Descriptor Detail	AGG_DESC_DET		A	
Aggregate Descriptor Item	AGG_DESC_ITEM		A	
Aggregate Organization	AGG_ORG		A	
Aggregate Purpose	AGG_PURP		A	
Aid Overpayment	AID_OVRPMT		A	
Collection	COLL	*	A	O
Deferment Type	DFR_TYPE		A	O
Expected Student Enrollment	EXP_STU_ENRL	*	A	O
FDLP Servicer	FDSLP_SVR		A	O
FDLP Servicer Branch Holder	FDSLP_SVR_BR_HOL	*	A	O
FDLP Servicer Submittal History	FS_SBMTL_HIS		A	
FDLP Servicer Submittal Run Error	FS_SBMTL_RUN_ERROR		A	
Financial Profile	FINANC_PROF	*	A	O
Guaranty Agency	GA		A	O
GA Aggregate	GA_AGG		A	
GA Submittal History	GA_SBMTL_HIS		A	
GA Submittal Run Errors	GA_SBMTL_RUN_ERR	*	A	
GA Summary	GA_SUM		A	
Insurance Claim Payment	INSUR_CL_PMT	*	A	O
Insurance Claim Refund	INSUR_CL_RFD		A	O
IRS Offset	IRS_OFF		A	O
Lender	LEN	*	A	O
Lender Branch	LEN_BR		A	O
Lender Branch Aggregate	LEN_BR_AGG	*	A	
Lender Branch Holder	LEN_BR_HOL	*	A	O
Lender Branch Holder Servicer	LEN_BR_HOL_SVR	*	A	O
Lender Branch Servicer	LEN_BR_SVR		A	O
Lender Branch Servicer Aggregate	LEN_BR_SVR_AGG		A	
Lender Branch Transition History	LEN_BR_TRAN_HIS		A	O
Loan	LOAN	*	A	O
Loan Cancellation	LOAN_CAN	*	A	O
Loan Deferment	LOAN_DFR	*	A	O
Loan Disbursement	LOAN_DIS	*	A	O
Loan External Identification	LOAN_EXTL_ID	*	A	
Loan Grouping	LOAN_GP	*	A	O
Loan Guarantor	LOAN_GUA	*	A	O

Table Description	Table Name	High Volume	Database	
Loan Maturity Date History	LOAN_MAT_DT_HIS	*	A	O
Loan Refund	LOAN_RFD		A	O
Loan Repayment Plan	LOAN_RPMT_PLAN	*	A	O
Loan Status	LOAN_STAT	*	A	O
Loan Supplement	LOAN_SUPP	*	A	
Loan Type	LOAN_TYPE		A	O
Loan Type Group	LOAN_TYPE_GP		A	
National Aggregate	NATL_AGG		A	
Output Distribution	OPUT_DISTR		A	
Organization Contact	ORG_CON		A	
PCA SPA Notification	PCA_SPA_NOTIF	*	A	O
Pell Grant	PELL_GRT	*	A	O
PEPS Submittal History	PEPS_SBMTL_HIS		A	
PEPS Submittal Run Error	PEPS_SBMTL_RUN_ERR		A	
PLUS Borrower	PLUS_BOR		A	O
PLUS Borrower Loan	PLUS_BOR_LOAN		A	O
PLUS Borrower Name	PLUS_BOR_NM		A	O
PLUS Borrower Social Security Number	PLUS_BOR_SSN		A	O
Preferred School	PREF_SCH	*	A	O
Prescreening Applicant	PRSCRN_APPL	*	A	
Prescreening Result	PRSCRN_RSLT	*	A	
Prescreening Result Loan	PRSCRN_RSLT_LOAN	*	A	
Prescreening Result Pell	PRSCRN_RSLT_PELL		A	
Region Aggregate	REG_AGG		A	
Reinsurance Claim Payment	REINSUR_CL_PMT	*	A	O
Reinsurance Claim Refund	REINSUR_CL_RFD		A	O
Repurchased Loan	RPCH_LOAN		A	O
Rate	RT		A	
School	SCH		A	O
School Branch	SCH_BR		A	O
School Branch Aggregate	SCH_BR_AGG	*	A	
School Branch Code Cross Reference	SCH_BR_CODE_XREF		A	
School Branch Holder Servicer	SCH_BR_HOL_SVR	*	A	O
School Branch Status History	SCH_BR_STAT_HIS		A	
School Branch Servicer	SCH_BR_SVR		A	O
School Branch Servicer Aggregate	SCH_BR_SVR_AGG		A	
School Branch Transmittal History	SCH_BR_TRAN_HIS		A	O
School Origination History	SCH_ORIGN_HIS		A	O
School Submittal History	SCH_SBMTL_HIS		A	
School Submittal Run Errors	SCH_SBMTL_RUN_ERR	*	A	
SPA Payment	SPA_PMT		A	O
SSCR Cycle	SSCR_CYCLE		A	
SSCR Return History	SSCR_RTN_HIS		A	
SSCR Transmittal Detail	SSCR_TRSMTL_DET	*	A	

Table Description	Table Name	High Volume	Database	
SSCR Transmittal History	SSCR_TRSMTL_HIS		A	
State Aggregate	ST_AGG		A	
Student	STU	*	A	O
Student Address	STU_ADD	*	A	
Student Branch	STU_BR	*	A	O
Student Demographics	STU_DEM	*	A	O
Student Name	STU_NM	*	A	O
Student Social Security Number	STU_SSN	*	A	O
Student Status	STU_STAT	*	A	O
Student Status Unresolved	STU_STAT_UNRESLV	*	A	
Supplemental Reinsurance Payment	SUPP_REINSUR_PMT	*	A	O
Validation Translation Table	VAL_TRSL_TAB		A	O

Legend: A=Active Database, O=Online Abstract, High Volume=In excess of 5 million records

Figure 4–17, Database Tables

The tables marked with an asterisk in Figure 5–4 contain over 5 million records. With these large volumes, it is particularly important to write efficient queries. To construct efficient queries, be sure to use indexes and qualify your queries with WHERE clauses.

4.4.4 Attribute Definitions Contained in NSLDS Tables

Section 4.4.3 provides a detailed listing of all database tables and their attributes, including the following information for each table:

- Table Name and Database Location (Active or Online Abstract)
- Table Description
- Attribute/Column Name
- Attribute/Column Description
- Data Type (Text, Numeric, Date, Timestamp, Time)
- Field Length
- Indicators of Pre-Established Indexes

Two examples of how to use Section 4.4 to develop an ad hoc report are presented below.

Example 1: Build an Ad Hoc Report

- **Scenario**—Produce a listing of branches for the University of Education (OPE ID = 777777). The listing shall include the school name, school branch name, school branch code, school branch city, and school branch state. To fulfill this request, look up the attribute definitions for the School Branch (SCH_BR) table in Section 5–2 and build the ad hoc query statement in Figure 4–18.

- **Query**—The ad hoc query statement in Figure 4–18 would then be used to generate the report in Figure 4–19.

```
SELECT  SCH_CODE
, SCH_NAME
, NM
, CODE
, CITY
, ST
 FROM   SCH_BR
WHERE   SCH_CODE = '777777'
ORDER
     BY  SCH_CODE
, CODE
```

Figure 4–18, School Branch Ad Hoc Query

- **Report**—This query then results in the report shown in Figure 4–19.

SCH_CODE	SCH_NAME	NM	CODE	CITY	ST
777777	UNIVERSITY OF EDUCATION	UNIVERSITY OF ED – MAIN BRANCH	00	WASHINGTON	DC
777777	UNIVERSITY OF EDUCATION	UNIVERSITY OF ED - BRANCH 01	01	DALLAS	TX
777777	UNIVERSITY OF EDUCATION	UNIVERSITY OF ED - BRANCH 02	02	DENVER	CO
777777	UNIVERSITY OF EDUCATION	UNIVERSITY OF ED - BRANCH 03	03	TAMPA	FL
777777	UNIVERSITY OF EDUCATION	UNIVERSITY OF ED - BRANCH 04	04	SAINT PAUL	MN
777777	UNIVERSITY OF EDUCATION	UNIVERSITY OF ED - BRANCH 05	05	ATLANTA	GA

Figure 4–19, School Branch Report

This example illustrates how to select the necessary columns (attributes) of a table using Section 4.4. Notice in Section 4.4 how each column is provided with a definition for ease-of-use, each column's data type and length are listed, and there is an exact match (=) used on the first column of the two columns that compose the identifier (Primary Key/Index) for the School Branch (SCH_BR) table. The Online Abstract Database table could be accessed with the same ad hoc query statement simply by typing ***OSAP.*** in front of SCH_BR.

Proceeding a little deeper into the capability of the NSLDS databases and Section 4.4, the following example illustrates how to *join* tables to make a more complex ad hoc query:

Example 2: Ad Hoc Query Using Joined Tables

- **Scenario**—Produce a list of students attending Branch 01 of the University of Education. The student list must include each student's identifiers (student number and student sequence number), Social Security Number, last name, first name, middle initial, and date of birth. To fulfill this request, refer to the attribute definitions for the Student Branch (STU_BR) and Student (STU) tables in Section 4.4 to build the following ad hoc query statement.

- **Query**—The ad hoc query statement in Figure 4–20 is used to generate the report in Figure 4–21.

```
SELECT  B.STU_NO
, B.STU_SEQ_NO
, S.CURR_SSN
, S.CURR_LST
, S.MID_INIT
, S.DOB
 FROM   STU_BR  B
, STU     S
WHERE  B.SCH_CODE       = '777777'
   AND  B.SCH_BR_CODE = '01'
   AND  B.STU_NO          = S.NO
   AND  B.STU_SEQ_NO    = S.SEQ_NO
ORDER
     BY  B.STU_NO
, B.STU_SEQ_NO
```

Figure 4–20, Join Student Branch and Student Ad Hoc Query

- **Report**—This query results in the report shown in Figure 4–21.

STU_NO	STU_SEQ_NO	CURR_SSN	CURR_LST	CURR_FST	MID_INIT	DOB
123456789	01	123-45-6789	ADAMS	TONY	A	1940-01-01
234567890	01	234-56-7890	CACHE	BILL	B	1941-01-01
345678901	01	345-67-8901	ALLEN	CHRIS	C	1950-12-31
456789012	01	456-78-9012	SPRUCE	JILL	D	1955-04-07
567890123	01	567-89-0123	MANWARING	GINA	N	1969-11-28
678901234	01	678-90-1234	LEACH	RICHARD	D	1969-12-07
901234567	01	901-23-4567	CHAMBERS	THAD	W	1963-11-29

NSL-0683

Figure 4–21, Join Student Branch and Student Report

This example shows how to join the appropriate columns (attributes) of the Student Branch (STU_BR) and Student (STU) tables. As in the first example, Section 4.4 provides the information necessary to build the ad hoc query statement. Notice that an exact match (=) was

used on both columns of the secondary index of the Student Branch (STU_BR) table. Also, an exact match (=) was used on both columns of the identifier (Primary Key/Index) for the Student (STU) table while *joining* to an exact match (=) of the first two of four columns that compose the identifier (Primary Key/Index) for the Student Branch (STU_BR) table. The Online Abstract Database tables can be accessed with the same ad hoc query statement simply by typing ***OSAP.*** in front of STU and STU_BR.

This ad hoc query can be expanded to retrieve loan-level data (such as current loan status, loan type, and outstanding principal balance) for each student, simply by joining the Loan (LOAN) table to the Student Branch (STU_BR) table using the *join* columns (STU_NO, STU_SEQ_NO, SCH_CODE, SCH_BR_CODE). You may have already discovered by looking at Section 4.4, that this *join* would use exact matches (=) on all four of the columns composing the secondary index of the Loan (LOAN) table and all four of the columns composing the identifier (Primary Key/Index) for the Student Branch (STU_BR) table.

These are just a few examples of what you can do by running ad hoc queries against the NSLDS database. To construct your own queries, refer to Chapters 6 for more information about how to formulate queries using QMF.

4.4.5 Table Relationships and Keys

The NSLDS database consists of many tables that are related to one another as parents and children are; therefore, they are called *parent tables* and *dependent tables*. These relationships are defined by keys (primary and foreign) that point from one table to another. Primary and foreign keys for all tables available to users are presented in Section 4.4.5.2. Using primary and foreign keys properly will allow you to search the enormous amount of data stored in NSLDS more quickly.

4.4.5.1 Parent Tables and Primary Keys

The most significant characteristic of the parent table is its primary key. The primary key (identifier) is constructed from one or more columns (attributes) of the parent table. The primary key is a pointer to one unique row (occurrence) in the parent table. The sequence of the columns that compose the primary key is very important. A primary key ALWAYS has an index defined upon its column(s) in the same exact order. In addition, the more columns of a primary key that are supplied an *equals to* (=) *value* in the key sequence, the quicker and more efficiently the query produces results from the table.

4.4.5.2 Dependent Tables and Foreign Keys

The dependent (child) table has a unique characteristic—a foreign key. A foreign key is a pointer to the primary key of the parent table. The columns of the foreign key match one-for-one the

columns of the primary key of the parent table. The column names may be spelled differently, but the column characteristic must be identical. Column characteristics are the type (such as character, integer, date, or time), length, and scale of the data in the column. The dependent table foreign key may exist one or more times in the dependent table, but always has one—and only one—parent row (primary key) in the parent table.

4.4.5.3 Table Relationships

The relationship between a parent table and a dependent table is just like a familial relationship. A parent can have one or more children, and every child has a parent. A parent table can be a dependent to another parent table. A dependent table can be the parent to another dependent table. Therefore, these relationships are consistent for grandparents through grandchildren.

4.4.6 Table Indexes

When writing queries to access various tables (entities) and particular columns (attributes) in those tables, you are directing the computer to search the entire database to retrieve specific information. This process is time consuming and expensive in terms of computer resources. To search and retrieve more efficiently, indexes were created to reduce the search effort. Section 4.4.6 lists the indexes available for writing queries. These indexes are also marked in Section 4.4.5 to show which tables and attributes have indexes, and which are primary and secondary. Also, the numbering indicates the sequence order of the columns (attributes) in the index.

An index is constructed of one or more column(s) from the associated table. Indexes are either *unique* or *non-unique*. A unique index must ALWAYS exist for a primary key (see Section 4.4.5); it serves as a pointer to one unique row (occurrence) of the associated table. Likewise, a non-unique index is a pointer to one or more row(s) of the associated table. Figure 4–22 shows three selected tables (center box) in juxtaposition with their related unique indexes (left box) and with a single non-unique index (right box). These unique indexes would clearly point to one and only one occurrence in the associated table, but the non-unique index (School Code and School Branch Code) could point to many occurrences in the Student Branch table.

Unique Index—Column List	
Index Name	**Column Name**
SYAGDI01	ID
SYAGDDI1	DESC_ID ID
STSTBRI1	STU_NO STU_SEQ_NO SCH_CODE SCH_BR_CODE

Table—Column List	
Table Name	**Column Name**
AGG_DESC	ID AGGR_NM FREQ
AGG_DESC_DET	ID DESC_ID
STU_BR	SYS_CR_DT ANTIC_CDMP_DT STU_NO VAL_IND STU_SEQ_NO SCH_CODE SCH_BR_CODE

Non-Unique Index—Column List	
Index Name	**Column Name**
none	
none	
STSTBRI2	SCH_CODE SCH_BR_CODE

Figure 4–22, NSLDS Tables Relative to Unique and Non-Unique Indexes

The sequence of the columns that compose an index is very important. In addition, the more columns of the index that can be supplied with an *equals to* (=) *value* in key-sequence order, in a search request, the faster the answer is retrieved from the table.

Example: Ad Hoc Query Using Indexes—If a table has an index of 3 columns (COL1-COL3) for a table of 5 million rows, retrieval of data with a search request of COL1=?, COL2=?, and COL3=? is much faster than a search specifying only one column (COL1 = X) or two columns (COL1=X and COL2=Y), because the search for the data is much more constrained or focused.

This comparison is even more pertinent if the requester has selected COL6=?, because there is no index on COL6. In this case all 5 million rows are searched. Also, a search request from the 5 million row table with COL1 >=? (or <=, or >=, etc.), COL2=?, and COL3=? is slow because COL1 >=? is going to search a portion of both the table and index. While this query is better than not using an index, it is not the most efficient use of the index.

Effective indexes are illustrated below:

1. An index on a table is similar to the index of a textbook or encyclopedia. If you are looking for the page (unique)—or pages (non-unique)—to find the definition of a certain word within the book, you could read quickly or search the book looking for sections where the word is used. But, you would probably do better using the index (which is in alphabetical order) to locate the word, retrieve the number(s) of the page(s) on which it is used, then go to the page(s) to find the definition of the word. It is the same when you ask the computer to find something. Although a computer can read extremely fast, it is still much more efficient to use an index.

2. Another example of an index is the card catalog system in your local library. If you are looking for a particular book, you go to the library card catalog (index), look up the location of the book using its author and title or Library of Congress cataloging number, and then go retrieve the book from the appropriate shelf location identified by the catalog.

To see the importance of the column sequence of an index, just imagine the same search for the library book assuming you only had the author's name. Even though you have the library card catalog, searching for and retrieving the book could be very time consuming because you might have to go through all of the cards under that author's name in the catalog looking for the book title you seek. Once you find the card with the matching author's name and book title, you only have to go to the shelf to locate the book. So supplying the author's name as the first part of the search is useful, but knowing the title enables you to find the book faster.

All of these principles are the same when you want to direct the computer to retrieve data. The more request values with an exact match, *equals to* (=), of the index columns in key sequence, the faster and more efficiently the computer can retrieve your data. Knowing the Library of Congress cataloging number for a book is like knowing a primary key (identifier) value for a table—it enables you to go right to where it is.

4.4.7 System Hierarchy

The Batch table and figures that follow identify the individual programs and screens, which make up the NSLDS and depict their logical and hierarchical organization.

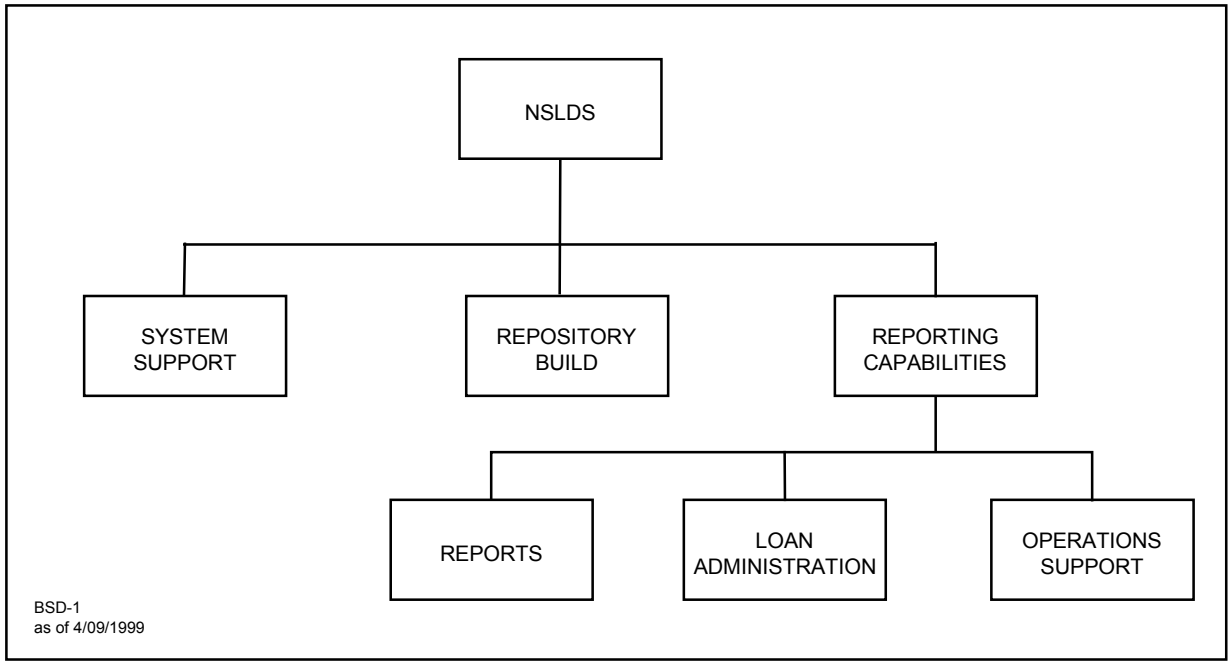

Figure 4–23, Business System Diagram, High Level

(Replace with **foldout** NSL-0835.flo)

Figure 4–24, NSLDS Batch Program Architecture

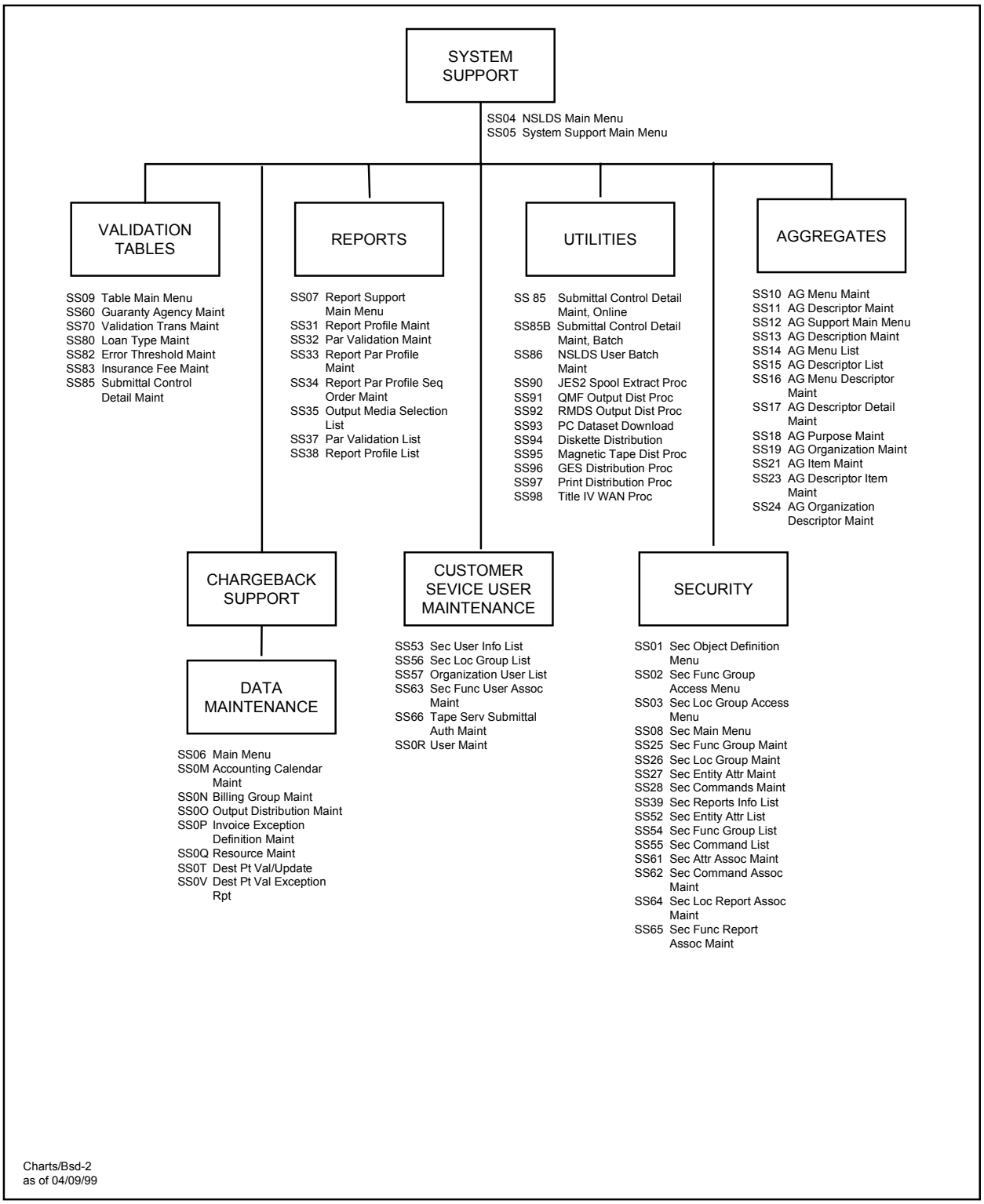

Figure 4–25, Business System Diagram, System Support

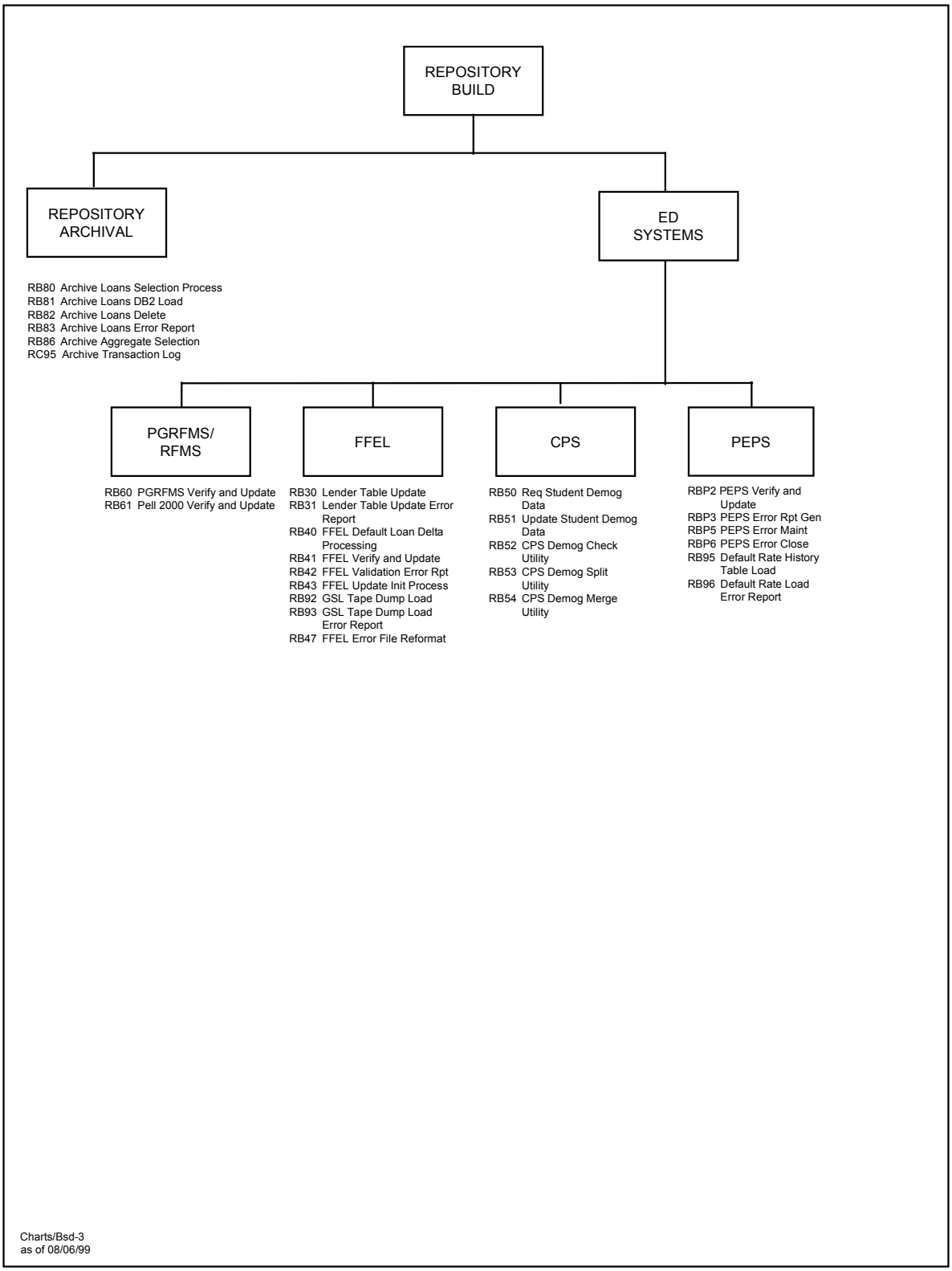

Figure 4–26, Business System Diagram, Repository Build 1: Repository Archival and ED Systems

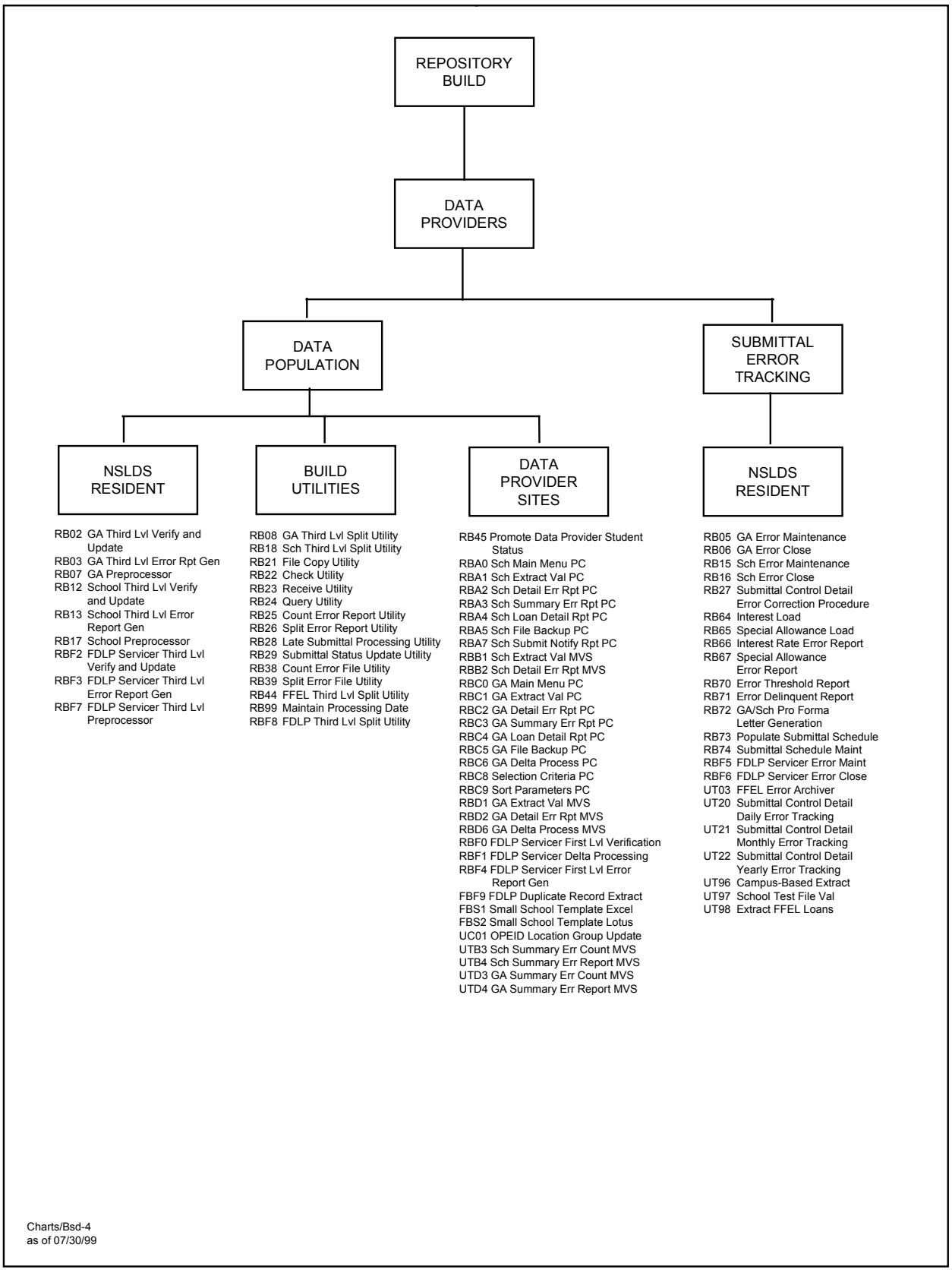

Figure 4–27, Business System Diagram, Repository Build 2: Data Providers

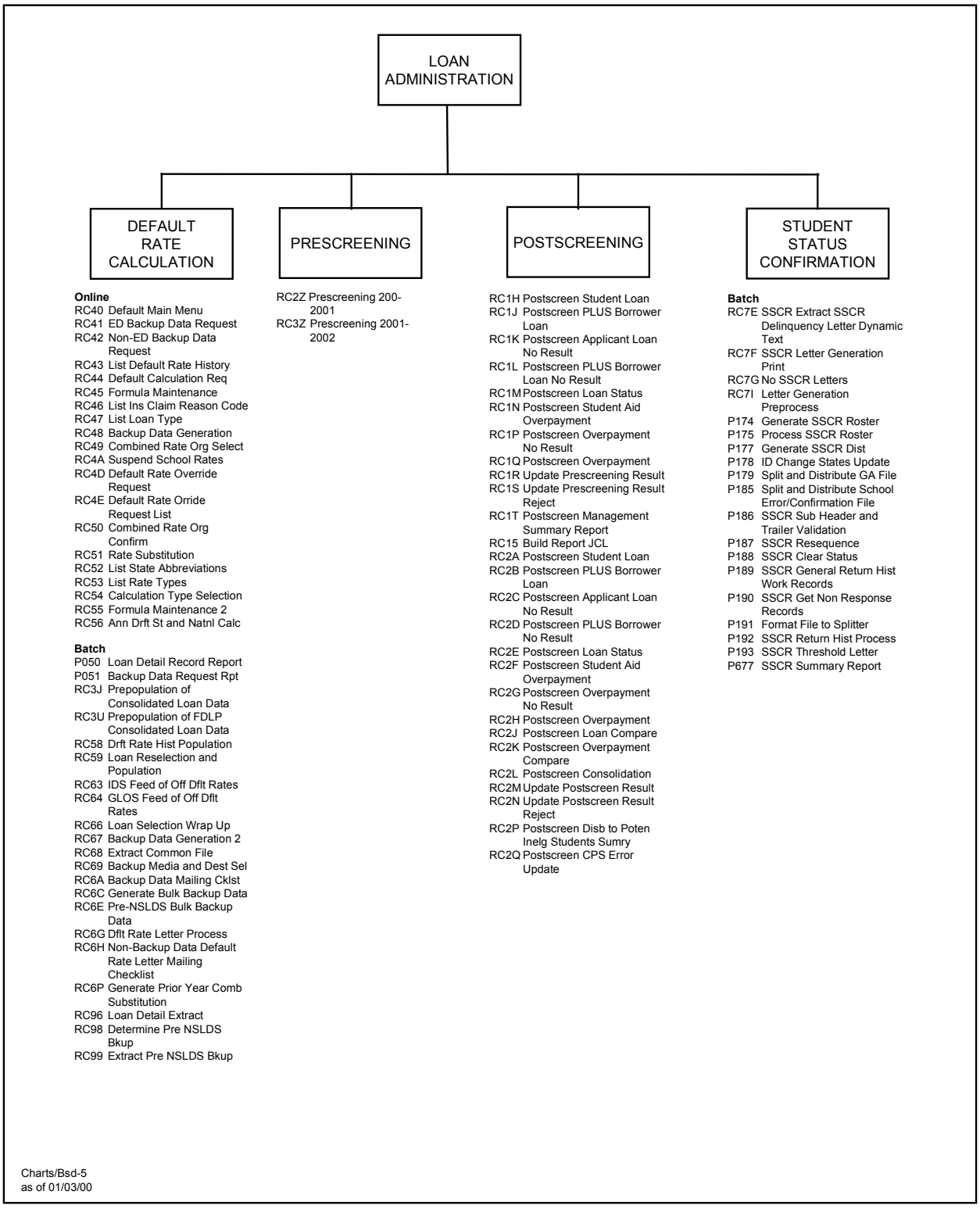

Figure 4–28, Business System Diagram, Reporting Capabilities 1: Loan Administration

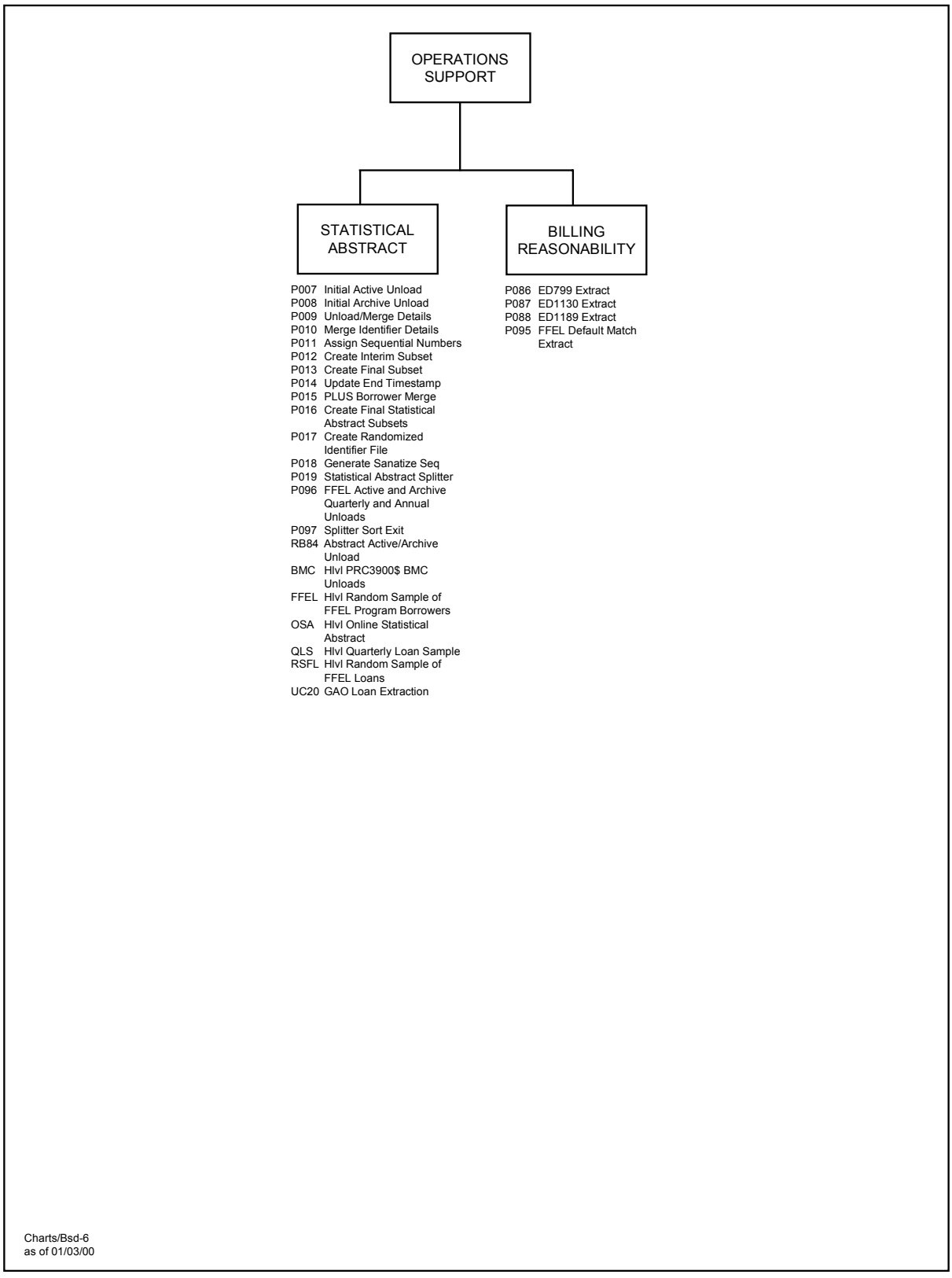

Figure 4–29, Business System Diagram, Reporting Capabilities 2: Operations Support

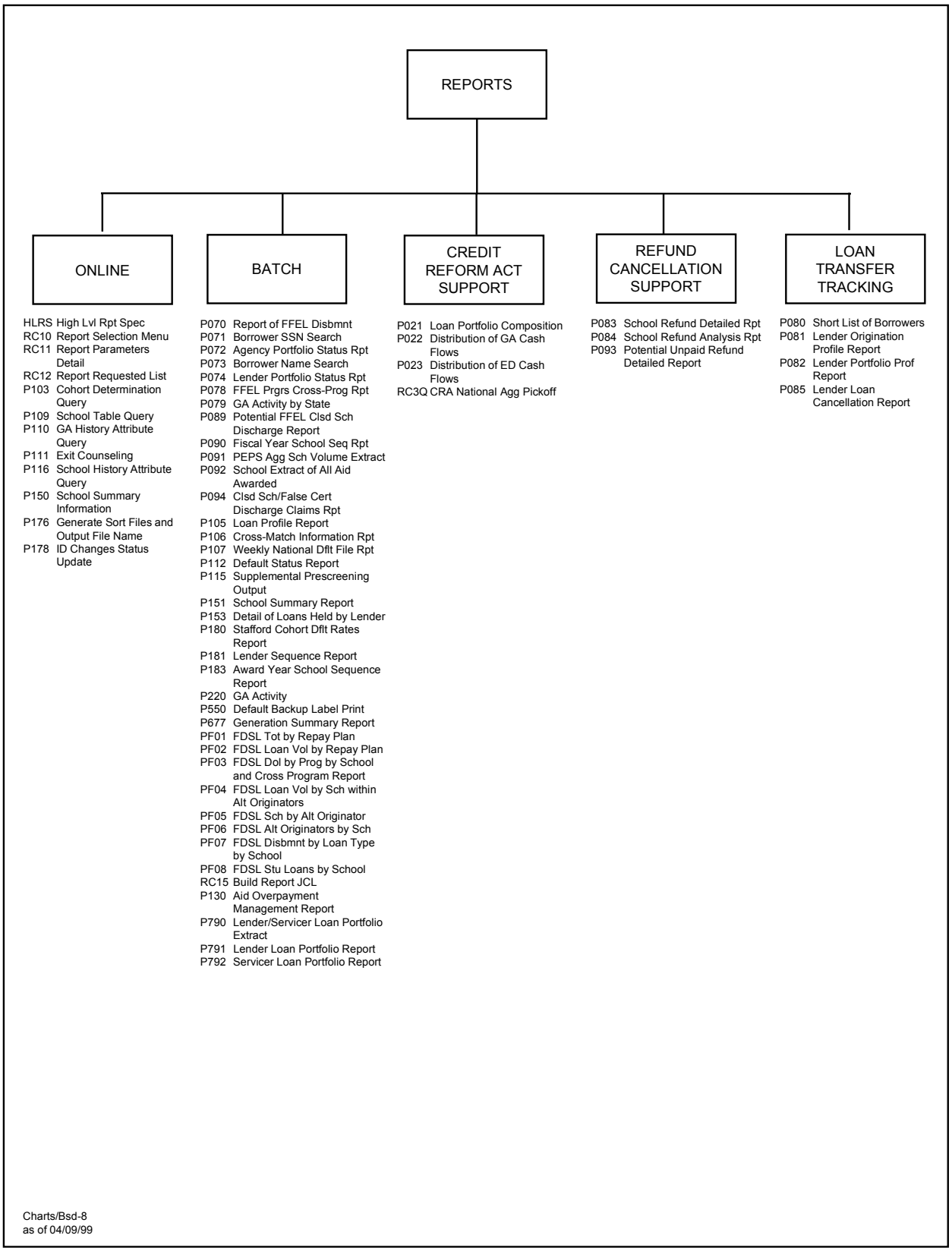

Figure 4–30, Business System Diagram, Reporting Capabilities 3: Reports

4.4.8 Data Model

The NSLDS data model serves as a broad gauge of what information is available in the system and is, therefore, a valuable reference point for an experienced user. This section presents the data model and its seven subject areas so users will better understand the basic structure of the NSLDS and thus be able to use its system capabilities more efficiently.

The NSLDS data model was constructed using Sterling Software's COOL:Gen Computer Aided Software Engineering (CASE) tool, formerly Texas Instruments' Information Engineering Facility (IEF). The COOL:Gen data model is based on a hierarchy of objects. Higher level objects decompose into lower level objects; for example, a COOL:Gen model name decomposes into subject areas, and subject areas decompose into entities and their relationships. The data model groups subject areas, and subject areas group closely related entities. Entities are collections of business data on which business functions operate. Data model diagrams depict relationships as lines that connect entities. Model names, subject areas, and entities are depicted as squares.

Business analysts use COOL:Gen's data model diagrams to check the logical consistency of a database, while database designers use them to produce a physical database using DB2, ORACLE, or a similar product. The NSLDS data model encompasses the seven main subject areas described in Figure 4–31.

Subject Area	Description
Students	Individuals whose education or partial education is funded through Title IV aid programs.
Schools	Institutions of higher education that participate in Title IV aid programs.
Lenders	Financial institutions that lend money under Title IV aid programs.
Guaranty Agencies	Agencies that guarantee Title IV loans for lenders. These agencies also aid in the administration of collections of defaulted loans, reinsurance of defaulted loans, and disbursement of Supplemental Preclaim Assistance for the FFEL program.
Loans	Money borrowed for funding a pre-approved educational program including disbursements, cancellations, deferments, refunds, loan aggregates, and uncollectables. These do not include grants.
FDLP Servicer	Organization that services FDLP loans for ED.
Default Rates	The creation of default rates for lenders, schools, and GAs related to the organizations' loans in the FFEL program.

Figure 4–31, Data Model Subject Areas

As you review the data model and the subject area diagrams, observe the overall structure of the system and the relationships between individual objects. This background is useful in determining whether the information you need is available on the system and how best to access it.

The following paragraphs briefly describe how to read and interpret the graphic presentations, using extracts from the School and Student subject areas as examples. The SCHOOL entity in Figure 4–32 has many attributes, including name, number, and address. SCHOOL BRANCH is another entity type that is related to SCHOOL in a *parent/child* relationship. It has many attributes, including Branch Code, branch name, and so forth. It is in a *child* relationship with SCHOOL because many SCHOOL BRANCHes may exist for a given SCHOOL, and these branches' existence depends on the SCHOOL entity. This is pictured in Figure 4–32 by the line that connects SCHOOL to SCHOOL BRANCH.

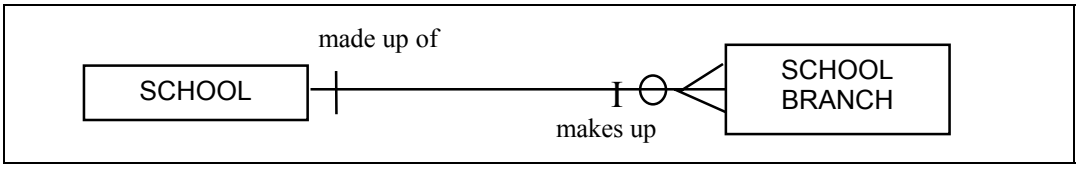

Figure 4–32, School and School Branch Entity Relationship Diagram

The line between SCHOOL and SCHOOL BRANCH also represents a *one-to-many* relationship; the crow's foot symbol on the SCHOOL BRANCH side of the line is the *many* symbol. This indicates that a SCHOOL may have many SCHOOL BRANCHes. The text and other symbols on the line describe the relationship in more detail. From the SCHOOL entity perspective, a SCHOOL is "made up of" zero, one, or more SCHOOL BRANCHes. The *O-symbol* indicates that the relationship is optional, and that a SCHOOL is not required to have a SCHOOL BRANCH.

The *I-symbol* on the relationship line indicates an *identifying relationship*. Identification of a SCHOOL BRANCH requires identifying elements from a SCHOOL. From the SCHOOL BRANCH perspective, a SCHOOL BRANCH "makes up" one SCHOOL. The vertical bar (|) on the SCHOOL end of the relationship indicates that the relationship is mandatory and that a SCHOOL BRANCH cannot exist without a SCHOOL.

Figure 4–33 defines the symbols displayed on connecting lines between entities in a data model:

Symbol	Name	Relationship	Indicates
|	vertical bar	one	Relationship is mandatory.
O	zero	optional	Relationship is optional.
<	crow's foot	many	The entity closest to this symbol may have multiple occurrences.
I	I	identifier	Requires identifying elements from both entities.

Figure 4–33, Data Model Symbols

SCHOOL and SCHOOL BRANCH represent actual organizations that exist in reality. Sometimes, however, an entity that does not have a corresponding existence in reality has to be created to track information. For example, a *many-to-many* relationship between SCHOOL BRANCH and STUDENT exists because a SCHOOL BRANCH can have many STUDENTs

attending and a STUDENT can attend many SCHOOL BRANCHes. *Many-to-many* relationships are not allowed in relational databases such as ORACLE or DB2. Therefore, *many-to-many* entity relationships are transformed creating a new associative entity. The STUDENT BRANCH associative entity contains values from the SCHOOL BRANCH entity and values from the STUDENT entity to show which SCHOOL BRANCH a STUDENT is attending. This is the case with the STUDENT BRANCH entity in Figure 4–34 where a STUDENT BRANCH associative entity was created to eliminate or resolve the *many-to-many* relationship.

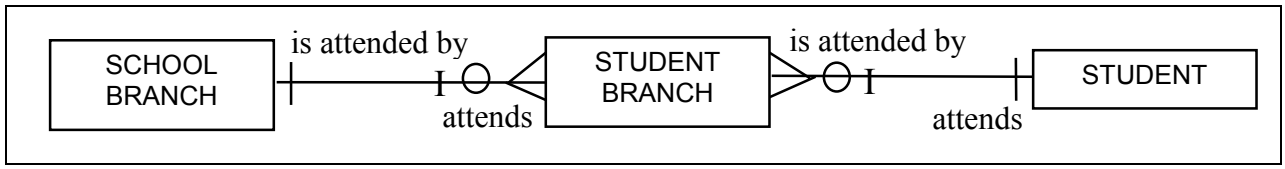

Figure 4–34, School Branch, Student Branch, and Student Entity Relationship Diagram

STUDENT BRANCH is known as an *associative* entity type because it tracks the association between a SCHOOL BRANCH and a STUDENT; the name is a derivation of the two entity types that are being associated. It is used as a cross-reference to track all the STUDENTs that attend a SCHOOL BRANCH as well as the different SCHOOL BRANCHes that a STUDENT may attend. This could be represented with a relationship line between SCHOOL BRANCH and STUDENT. However, important information must be tracked at this SCHOOL BRANCH / STUDENT intersection such as *expected graduation date* as well as the keys to identify a particular SCHOOL BRANCH that a particular STUDENT is attending. It would be incorrect to store this information at the SCHOOL BRANCH level or the STUDENT level since both a particular SCHOOL and a particular STUDENT must be known before *expected graduation date* can be recorded. For this reason, an associative entity type is used. It is also known as an *intersection* entity type and is quite common in other NSLDS subject areas.